

Iterator – IN1010

Hvorfor implementere en iterator?

Muliggjør enkel syntaks for å iterere gjennom en beholder. Det kan være en ArrayList, Lenkeliste e.l.

Hvordan implementere en iterator i en lenkeliste?

1. import java.util.Iterator

2. Lenkelisten må implementere interfacet Iterable<T>

3. Lag en indre klasse i klassen Lenkeliste:

```
class LenkelisteIterator implements Iterator<T> {
    private Node temp;

    public LenkelisteIterator(){
        temp = start;
    }

    @Override
    public boolean hasNext(){
        Så lenge det finnes flere Noder i lenken vår skal denne
        metoden returnere true. Det vil si så lenge «temp» ikke
        er null. «temp» settes i konstruktøren til å referere til
        det samme som lenkelistens «start», altså referansen til
        første Node i lenkelisten (hvis lenkelisten er tom er
        denne null).
    }

    @Override
    public T next(){
        Her skal vi returnere dataen (se «T» som returverdi) til
        gjeldende Node, altså «temp». Vi må også oppdatere
        «temp» til å referere til neste Node i lenken. Tanken er
        at vi alltid kaller hasNext() før vi kaller next(). Vi sjekker
        altså alltid om det faktisk finnes enda en Node før vi evt
        returnerer denne Nodens innhold.
    }
}
```

4. Lag en metode i klassen Lenkeliste:

```
public Iterator<T> iterator(){
    return new LenkelisteIterator();
}
```

Hvordan bruke den implementerte iteratoren?

Enhanced for loop: («Ting» erstattes med den typen listen består av)

```
for (Ting t : liste){  
    System.out.println(t)  
}
```

Bruke iteratorobjektet direkte:

```
Iterator iter = liste.iterator();  
while (iter.hasNext()){  
    System.out.println(iter.next());  
}
```